Random Walk Graph Laplacian based Smoothness Prior for Soft Decoding of JPEG Images

Xianming Liu

http://homepage.hit.edu.cn/pages/xmliu
http://arxiv.org/abs/1607.01895

ICME2016 Tutorial

Overview

- Background
- Popular Priors
  - Laplacian Prior
  - Sparsity Prior
  - Graph-signal Smoothness Prior
- Random Walk Graph Laplacian Regularizer
- Soft Decoding based on Priors Mixture
- Experimental Results
- Conclusion

Background

- Compressed image restoration: important and practical problem:
  - Compression is the most common cause of image degradation.
  - Compression is indispensable in almost all visual communication systems.

- Compressed image restoration is a non-trivial problem:
  - Compression noises are signal-dependent.
  - Far from being white and independent.
  - Composite noises: blocking and ringing effects.

JPEG Image Restoration

□ Problem Formulation

- **Encoder:**

DCT

$$q_i = round(\mathrm{Y}_i / \mathrm{Q}_i), \quad \mathbf{Y} = \mathbf{T}\mathbf{y}$$

8x8 pixel block

quantization parameter

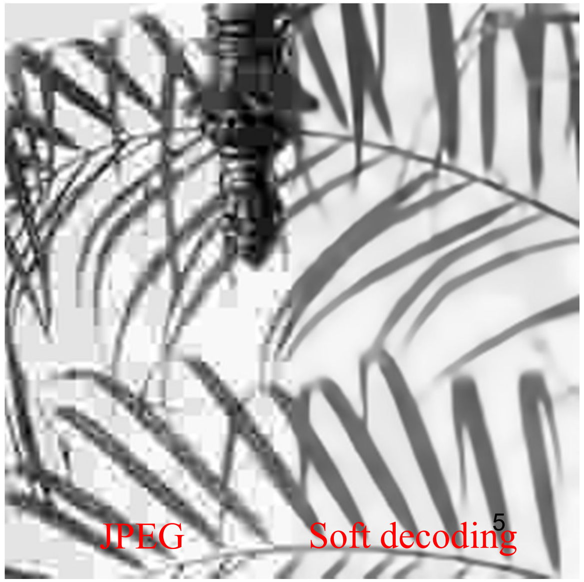
DCT Coefficients

- **Decoder: the quantization bin (q-bin) constraint**

$$q_i \mathrm{Q}_i \leq \mathrm{Y}_i \leq (q_i + 1)\mathrm{Q}_i, i = 1, 2, \cdots, 64.$$

Hard Decoding vs. Soft Decoding

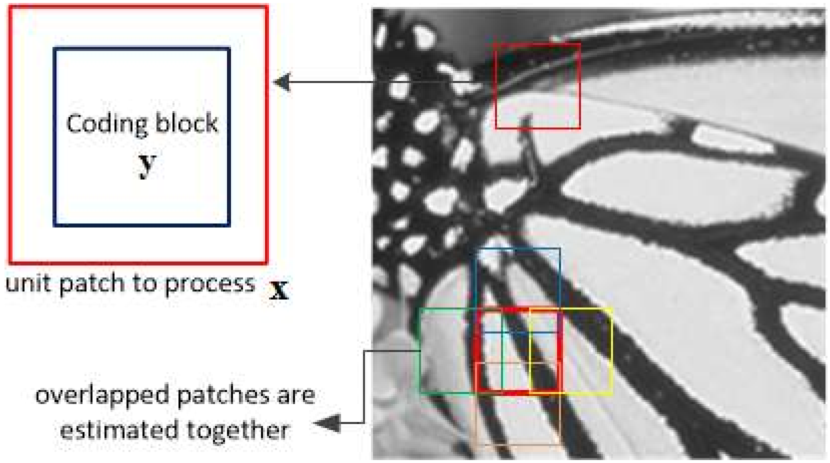
- Hard Decoding
  - Reconstruct DCT coefficients using the centers of assigned quantization bins.
- Soft Decoding
  - Find the most probable signal WITHIN the set of quantization bin constraints.
  - Signal priors is used for aid
    - Laplacian [Lam and Goodman, TIP'00]
    - Local/non-local similarity [Zakhor, TCSVT'92] [Zhai et al., TCSVT'08, TMM'08] [Zhang et al., TIP'14]
    - Total Variation [Bredies, SIAM J. Img. Sci'12]
    - Sparsity [Jung et al., SPIC'12] [Liu et al., CVPR'15, TIP'16]
    - Sparsity + TV [Chang et al. TSP'15]
    - Low-rank Prior [Zhao et al., TCSVT'16][Zhang et al, TIP'16]



Related Work of Graph-based Image Restoration and Enhancement

- Denoising [Hu et al., MMSP'14, ICIP'14], [Pang et al. APSIPA'14, ICASSP'15]
- Super-resolution [Mao et al., GlobalSIP'13, 3DTV'14]
- Dequantization [Liu et al, ICIP'15][Hu et al.,SPL'16]
- Deblurring [Kheradmand and Milanfar, TIP'14]
- Bit-depth Enhancement [Wan et al., TIP'16]
- Joint Denoising and Contrast Enhancement [Liu et al., ICASSP'15]

MAP Formulation



$$\mathbf{y} = \mathbf{M}\mathbf{x}$$

- patch surrounds block
- **x** is the basic processing unit

- Maximum a posterior (MAP):

$$\begin{aligned}\mathbf{x}^* &= \arg\max_{\mathbf{x}} p\left(\mathbf{x} \mid \mathbf{q}\right) \\ &= \arg\max_{\mathbf{x}} p\left(\mathbf{q} \mid \mathbf{x}\right) p(\mathbf{x}).\end{aligned}$$

- The likelihood is defined as:

$$p(\mathbf{q} \mid \mathbf{x}) = \begin{cases} 1 & \text{if } \mathrm{round}(\mathbf{TMx}/\mathbf{Q}) = \mathbf{q} \\ 0 & \text{o.w.} \end{cases}$$

- MAP formulation becomes

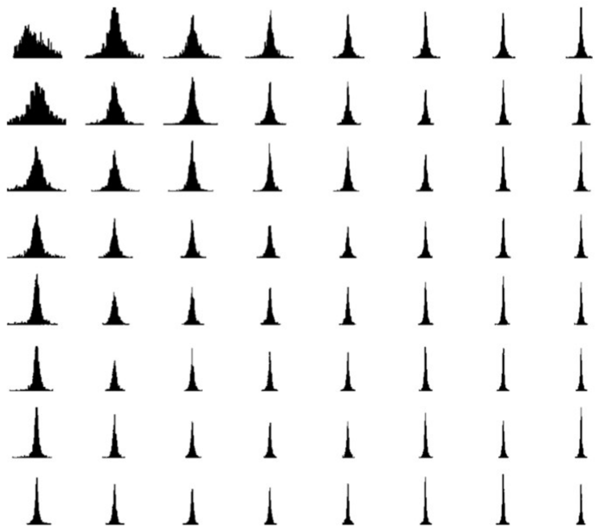
$$\begin{aligned}\mathbf{x}^* &= \arg\max_{\mathbf{x}} p(\mathbf{x}). \\ \text{s.t.} \quad & \mathbf{qQ} \preceq \mathbf{TMx} \prec (\mathbf{q}+1)\mathbf{Q}\end{aligned}$$

Laplacian Prior

- Q-bins: constrain the search space of individual DCT coefficients
- Laplacian Prior: states the probability density function of individual DCT coefficients

$$P_L(Y_i) = \frac{\mu_i}{2} \exp(-\mu_i |Y_i|)$$

[Lam and Goodman, TIP'00]



- MMSE Formulation

$$Y_i^* = \arg\min_{Y_i^o} \int_{q_i Q_i}^{(q_i+1)Q_i} (Y_i^o - Y_i)^2 \, P_L(Y_i) \, dY_i.$$

- Closed-form Solution

$$Y_i^* = \frac{(q_i Q_i + \mu_i)\, e^{\left\{\frac{-q_i Q_i}{\mu_i}\right\}} - ((q_i+1)Q_i + \mu_i)\, e^{\left\{\frac{-(q_i+1)Q_i}{\mu_i}\right\}}}{e^{\left\{\frac{-q_i Q_i}{\mu_i}\right\}} - e^{\left\{-\frac{(q_i+1)Q_i}{\mu_i}\right\}}}$$

For higher frequencies, the Laplacian parameter is larger; i.e., the distribution is sharper and more skewed to 0.

Laplacian Prior

- Advantage
  - closed-form MMSE solution
  - smaller expected squared error than a MAP solution

- Limitation
  - can only be used to recover code blocks separately
  - cannot handle block artifacts that occur across adjacent blocks

- Solution
  - We turn to employ the sparsity prior at a larger patch level **x**.

Sparsity Prior

Sparse Signal Model

$$\mathbf{x} = \boldsymbol{\Phi}\boldsymbol{\alpha} + \boldsymbol{\xi}$$

over-complete dictionary ($\boldsymbol{\Phi}$)   sparse code ($\boldsymbol{\alpha}$)

Sparse Coding

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{x} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_0,$$

- orthogonal matching pursuit (OMP) [Cai and Wang, TIT'11]
- computational complexity is linear with the size of dictionary

Sparsity Prior

$$P_S(\mathbf{x}) \propto \exp(-\lambda \|\boldsymbol{\alpha}\|_0).$$

Sparsity-based Soft Decoding

$$\min_{\{\mathbf{x},\boldsymbol{\alpha}\}} \|\mathbf{x} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_0 ,$$
$$\text{s.t. } \mathbf{qQ} \preceq \mathbf{TMx} \prec (\mathbf{q}+1)\mathbf{Q}$$

- *Step 1–Initial Estimation:* The Laplacian prior is used to get an initial estimation of **x**.
- *Step 2–Sparse Decomposition:*

$$\boldsymbol{\alpha}^{(t)} = \arg\min_{\boldsymbol{\alpha}} \left\|\mathbf{x}^{(t)} - \boldsymbol{\Phi}\boldsymbol{\alpha}\right\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_0 ,$$

- *Step 3–Quantization Constraint:*

$$\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x}} \left\|\mathbf{x} - \boldsymbol{\Phi}\boldsymbol{\alpha}^{(t)}\right\|_2^2 ,$$
$$\text{s.t. } \mathbf{qQ} \preceq \mathbf{TMx} \prec (\mathbf{q}+1)\mathbf{Q}$$

Lemma 1: The sparsity-based soft decoding algorithm converges to a local minimum.

Limitation of Small KSVD Dictionary

- Complexity linearly increases with the size of dictionary.
- In practice, a just reasonable over-complete dictionary is used.
- KSVD Dictionary Training

$$\min_{\boldsymbol{\Phi},\{\boldsymbol{\alpha}_i\}} \sum_{i=1}^{N} \|\mathbf{x}_i - \boldsymbol{\Phi}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_0,$$

Training pixel patch

DCT patch $\mathbf{X}_i = \mathbf{T}'\mathbf{x}_i$

Parsavel's theorem

pre-set sparsity limit

$$\min_{\boldsymbol{\Phi},\{\boldsymbol{\alpha}_i\}} \sum_{i=1}^{N} \|\mathbf{X}_i - \mathbf{T}'\boldsymbol{\Phi}\boldsymbol{\alpha}_i\|_2^2, \quad \text{s.t.,} \quad \|\boldsymbol{\alpha}_i\|_0 \le K$$

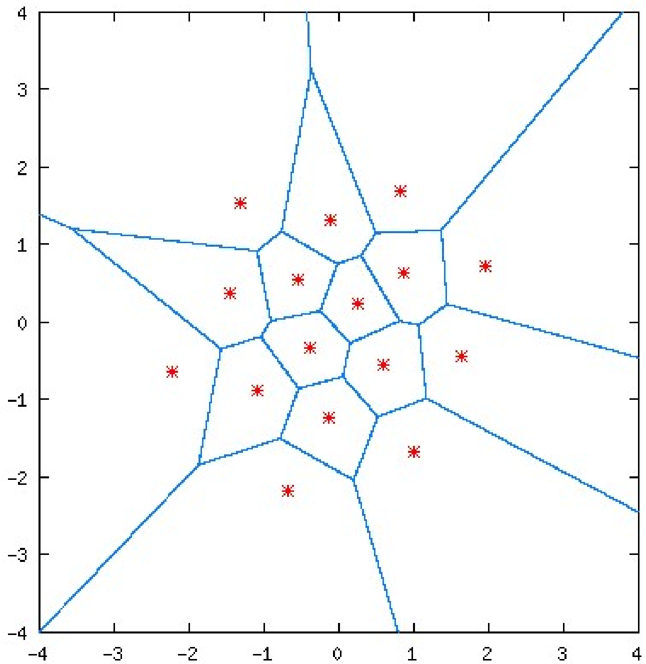
We analyze the behavior of dictionary learning in frequency domain

Limitation of Small KSVD Dictionary

When K = 1, dictionary learning becomes vector quantization (VQ) design problem

- Selecting *M* atoms is analogous to designing *M* partitions

$$\mathbf{R} = \cup_{m=1}^{M} \mathbf{R}_m \qquad \mathbf{R}_i \cap \mathbf{R}_j = \emptyset, \forall i \neq j$$



Limitation of Small KSVD Dictionary

When K = 1, dictionary learning becomes vector quantization (VQ) design problem

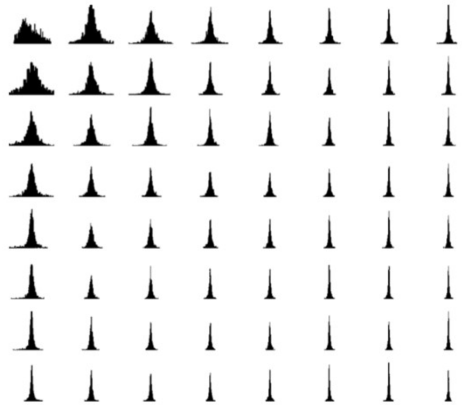
- Selecting *M* atoms is analogous to designing *M* partitions

$$\mathbf{R} = \cup_{m=1}^{M} \mathbf{R}_m \qquad \mathbf{R}_i \cap \mathbf{R}_j = \emptyset, \forall i \neq j$$

- When *N* tends to infinite:

$$\min_{\{\phi_m\}} \sum_{m=1}^{M} \int_{\mathbf{R}_m} \|\mathbf{X} - \mathbf{T}'\phi_m\|_2^2 P(\mathbf{X}) d\mathbf{X}$$

Expected square error



a product of Laplacian distributions for individual DCT frequencies

- **low frequencies: decay slowly**
- **high frequencies: more skewed and concentrated around zero**

Limitation of Small KSVD Dictionary

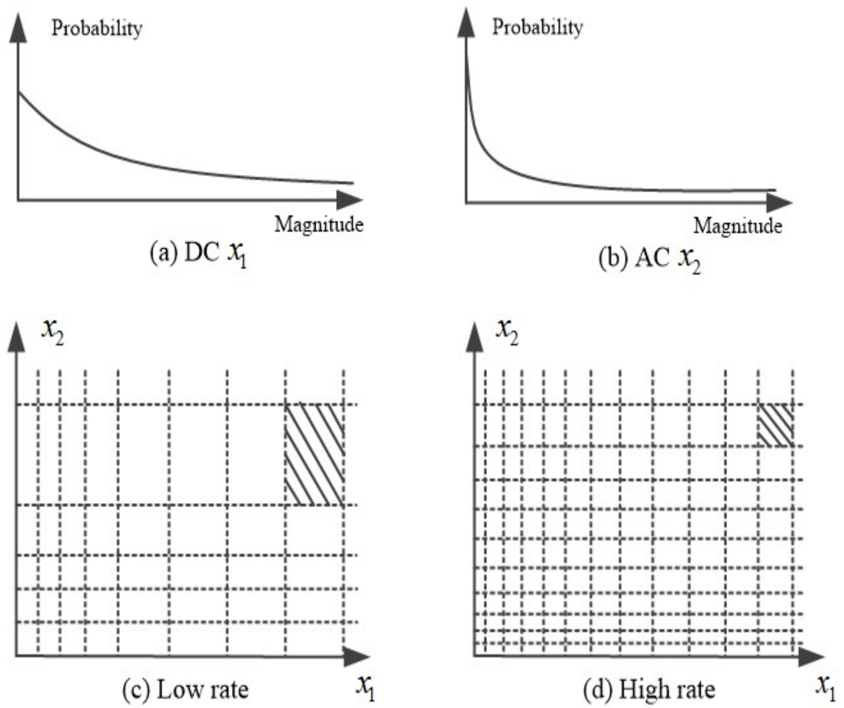


Illustration of product VQ for DC and AC frequencies

- When the number of atoms is small
  - quantization is coarser for large magnitude in AC than DC

When the dictionary Φ is small, the sparsity prior is difficult to recover large magnitude of high DCT frequencies.

- When the dictionary is large enough
  - quantization for large magnitude in high frequency is sufficiently fine.

When the dictionary Φ is large enough, the sparsity prior can recover large magnitude of high DCT frequencies well.

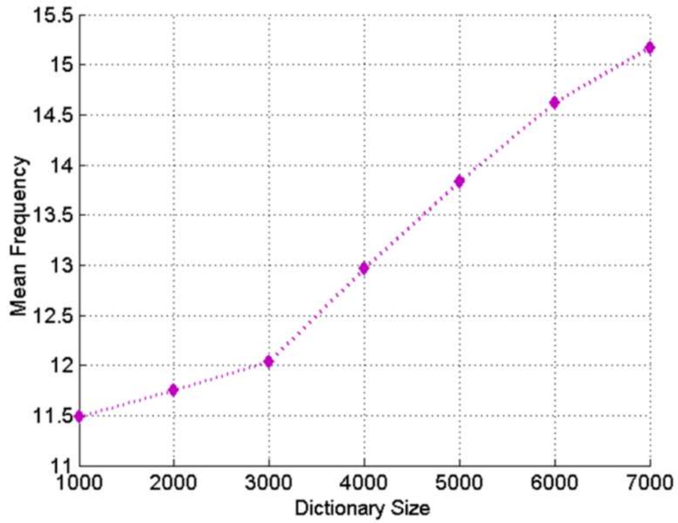
Empirical Observation

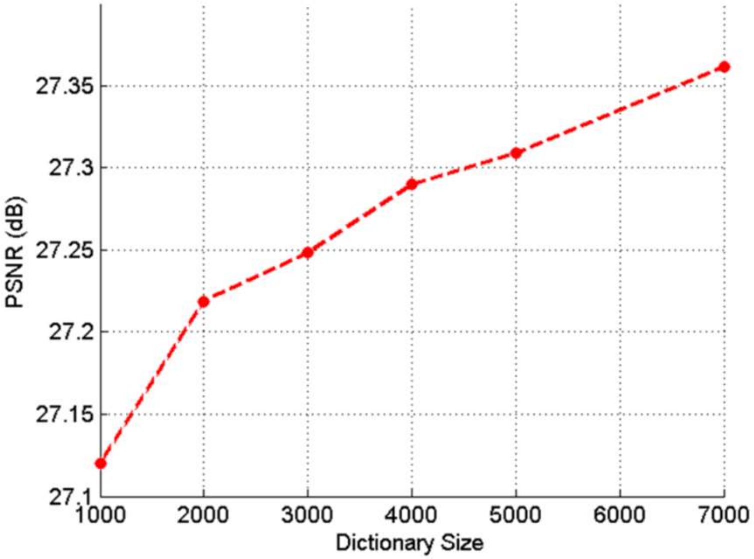
Mean Frequency

$$MF = \frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{n} f_i Y_i^2(m)$$

DCT frequency

DCT coefficient of atom





(a) Image1 (PSNR: 26.87, SSIM: 0.8982) (b) Image2 (PSNR: 27.30, SSIM: 0.9039)

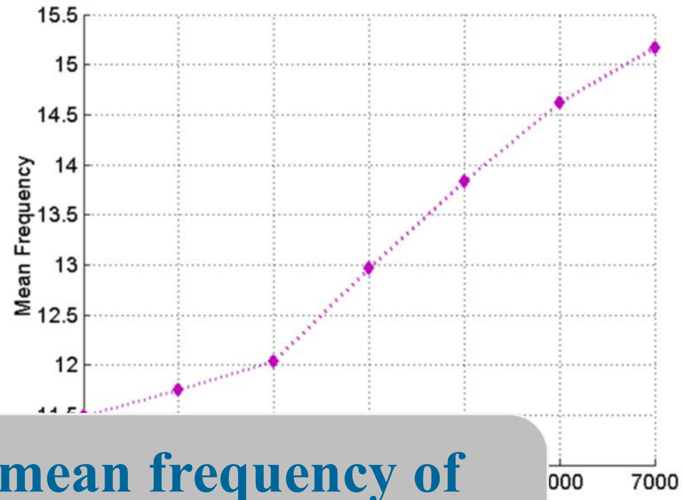
Empirical Observation

□ Mean Frequency

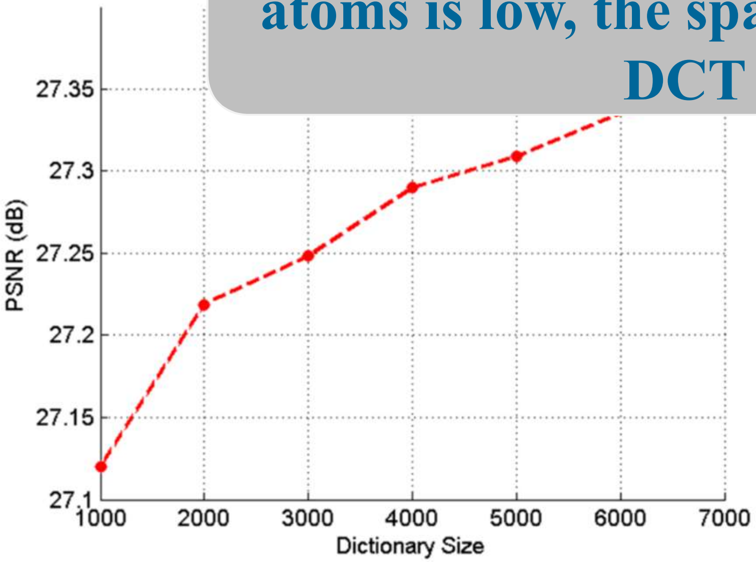
$$MF = \frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{n} f_i Y_i^2(m)$$

DCT f...

DCT coefficient

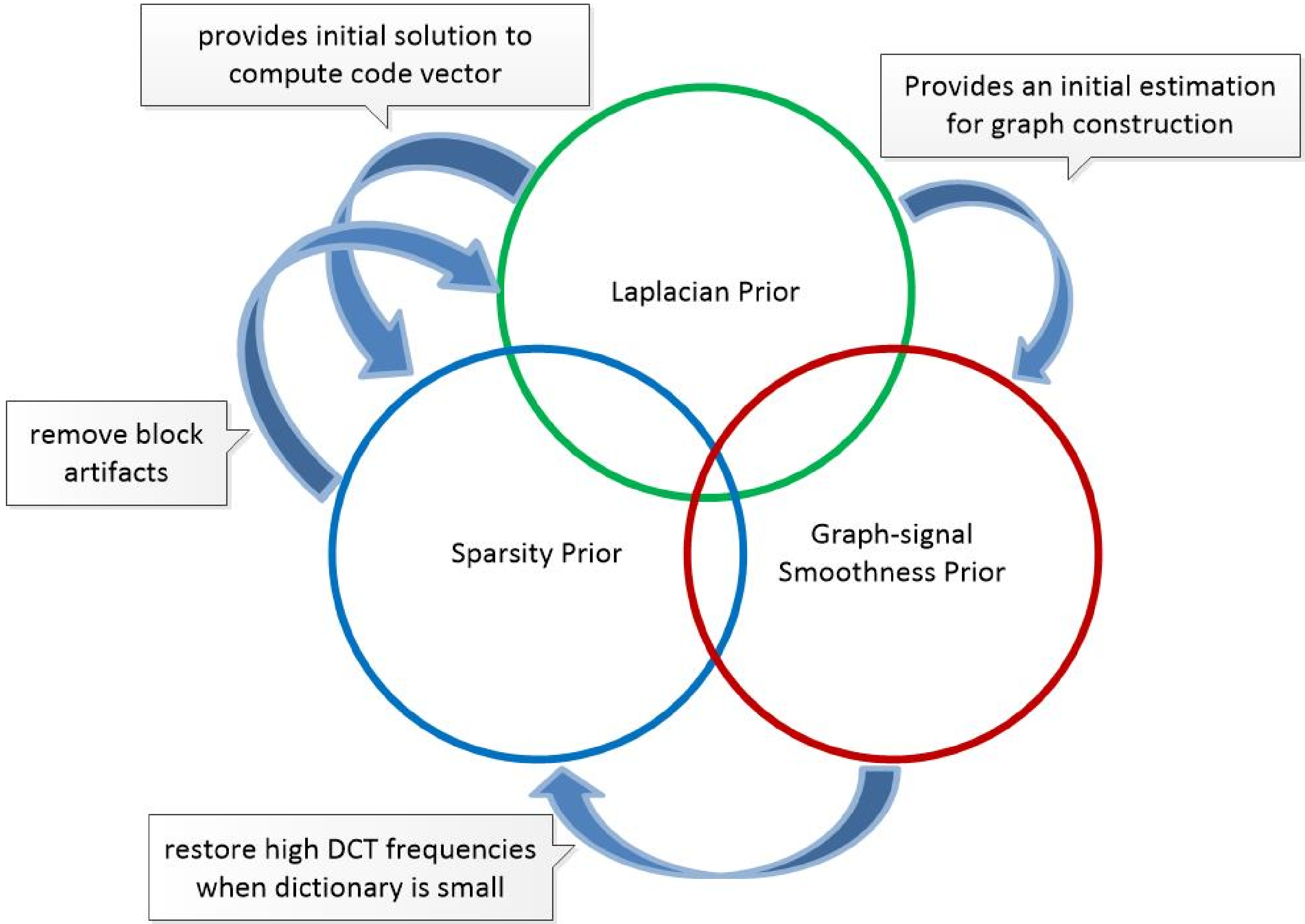


When dictionary is small, the mean frequency of atoms is low, the sparsity prior cannot recover high DCT frequencies well.



(a) Image1 (PSNR: 26.87, SSIM: 0.8982) (b) Image2 (PSNR: 27.30, SSIM: 0.9039)

Three Priors Complement Each Other



Graph-signal Smoothness Prior

Graph Laplacian Regularizer

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} (x_i - x_j)^2 W_{i,j} \quad \Longrightarrow \quad P_G(\mathbf{x}) \propto \exp\left(-\lambda_2 \mathbf{x}^T\mathbf{L}\mathbf{x}\right)$$

Different graph Laplacian matrixes

- Combinatorial graph Laplacian: $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- Symmetrically normalized graph Laplacian: $\mathcal{L}_n = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$
- Random walk graph Laplacian: $\mathcal{L}_r = \mathbf{D}^{-1}\mathbf{L}$
- Doubly stochastic graph Laplacian: $\mathcal{L}_d = \mathbf{I} - \mathbf{C}^{-1/2}\mathbf{W}\mathbf{C}^{-1/2}$

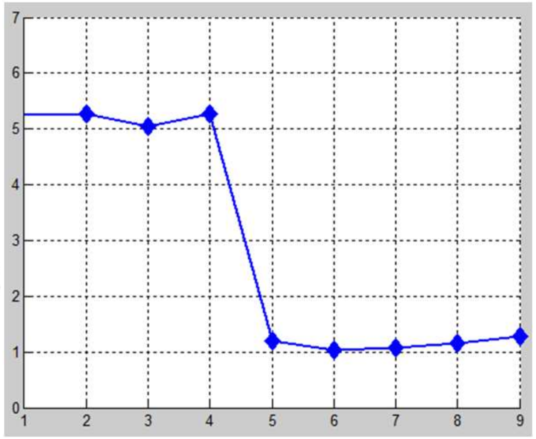
Graph Laplacian	Symmetric	DC eigenvector
Combinatorial	Yes	Yes
Symmetrically Normalized	Yes	No
Random Walk	No	Yes
Doubly Stochastic	Yes	Yes

[Kheradmand and Milanfar, TIP'14]

Graph-signal Smoothness Prior

- Graph Frequency Interpretation
  - Eigen decomposition: $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$
    - eigenvalues carry the notion of frequency
  - Graph Fourier transform: $\mathbf{F} = \mathbf{U}^T$ → $\boldsymbol{\alpha} = \mathbf{F}\mathbf{x}$
  - We get

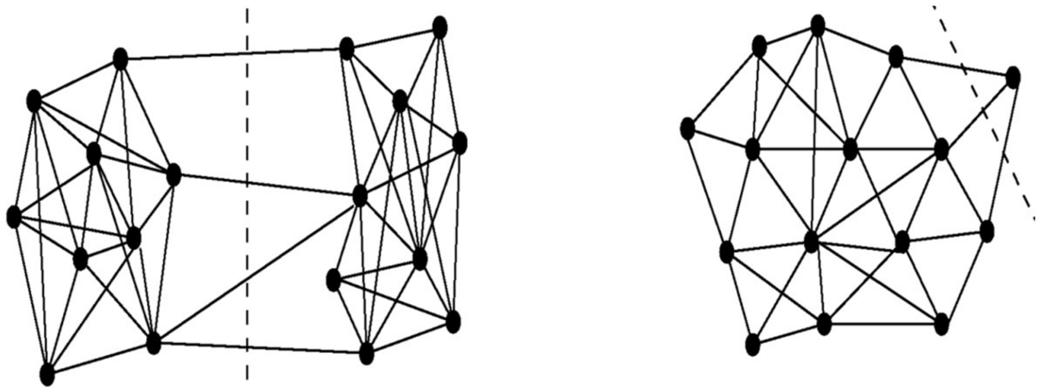
$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \boldsymbol{\alpha}^T\mathbf{\Lambda}\boldsymbol{\alpha} = \sum_k \eta_k \, \alpha_k^2.$$



- Minimizing $\mathbf{x}^T\mathbf{L}\mathbf{x}$ will suppress high graph frequencies and preserve low graph frequencies.
  - **x** is smoothened with respect to the graph
  - PWS signals can be well approximated by low graph frequencies for appropriately constructed graphs. [Hu et al., MMSP'14, ICIP'14]
  - Discontinuities inside PWS signals translate to high DCT frequencies.

Why Graph Prior Works Well for PWS Signals?

- Spectral clustering: given a similarity graph, separate its vertices into two subsets of roughly the same size via spectral graph analysis.
- Normalized cut (Ncut) [Shi and Malik, TPAMI'00]



NP-hard!

$$\min_{\mathcal{A},\mathcal{B}} \mathrm{Ncut}(\mathcal{A},\mathcal{B}) = \min_{\mathbf{f}} \frac{\mathbf{f}^T\mathbf{L}\mathbf{f}}{\mathbf{f}^T\mathbf{D}\mathbf{f}}$$

where

$$\mathbf{f} = [f_1, \cdots, f_n]^T \text{ and } f_i = \begin{cases} \frac{1}{\mathrm{vol}(\mathcal{A})} & \text{if } i \in \mathcal{A} \\ \frac{-1}{\mathrm{vol}(\mathcal{B})} & \text{if } i \in \mathcal{B} \end{cases}$$

PWC!

$$\min_{\mathbf{f}} \frac{\mathbf{f}^T\mathbf{L}\mathbf{f}}{\mathbf{f}^T\mathbf{D}\mathbf{f}}, \quad \text{s.t.} \quad \mathbf{f}^T\mathbf{D}\mathbf{1} = 0$$

$$\mathbf{v} := \mathbf{D}^{1/2}\mathbf{f}$$

$$\mathbf{v}_1 := \mathbf{D}^{1/2}\mathbf{1}$$

Relaxed solution of Ncut!

$$\min_{\mathbf{v}} \frac{\mathbf{v}^T\mathcal{L}_n\mathbf{v}}{\mathbf{v}^T\mathbf{v}}, \quad \text{s.t.} \quad \mathbf{v}^T\mathbf{v}_1 = 0$$

Interpretation from the Perspective of Spectral Clustering

Rayleigh quotient with respect to $\mathcal{L}_n$ ⟸ $$\min_{\mathbf{v}} \boxed{\frac{\mathbf{v}^T \mathcal{L}_n \mathbf{v}}{\mathbf{v}^T \mathbf{v}}}, \quad \text{s.t.} \quad \mathbf{v}^T \mathbf{v}_1 = 0$$

- v_1 minimizes the objective, since $\mathbf{v}_1^T \mathcal{L}_n \mathbf{v}_1 = \mathbf{1}^T \mathbf{L} \mathbf{1} = 0$
  - v_1 is the first eigenvector of $\mathcal{L}_n$
- v is orthogonal to v_1 , according to Rayleigh quotient, the solution is the second eigenvector of $\mathcal{L}_n$

> The second eigenvector v_2 of L_n is a relaxed solution to the Ncut problem, which is PWS;
> if the solution becomes exact, then v_2 is PWC.

- Low graph frequencies of $\mathcal{L}_n$ thus are suitable to compactly represent PWS signals.

Random Walk Graph Laplacian

- The first eigenvector of $\mathcal{L}_n$, $\mathbf{v}_1 := \mathbf{D}^{1/2}\mathbf{1}$, is not a constant vector → $\mathcal{L}_n$ does not have DC component → not suitable for filtering natural images.
- Matrix similarity transformation[1]

$$\mathcal{L}_r := \mathbf{D}^{-1/2}\mathcal{L}_n\mathbf{D}^{1/2} = \mathbf{D}^{-1}\mathbf{L}$$

→ **Random walk graph Laplacian!**

- $\mathcal{L}_r$ has the left eigenvectors $\mathbf{V}^T\mathbf{D}^{1/2}$

$$\mathbf{V}^T\mathbf{D}^{1/2}\mathcal{L}_r = \mathbf{\Lambda}\mathbf{V}^T\mathbf{D}^{1/2} \qquad \mathcal{L}_n = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

- GFT using the left eigenvectors

$$\boldsymbol{\beta} = \mathbf{V}^T\mathbf{D}^{1/2}\mathbf{x}$$

[1] https://en.wikipedia.org/wiki/Matrix_similarity

Random Walk Graph Laplacian

- However, $\mathcal{L}_r$ is asymmetric, there is no clear interpretation in graph frequency domain of $\mathbf{x}^T\mathcal{L}_r\mathbf{x}$.
- We use $\mathcal{L}_r^T\mathcal{L}_r$ instead, and can derive:

$$\mathbf{x}^T\mathcal{L}_r^T\mathcal{L}_r\mathbf{x} = (\mathbf{x}^T\mathbf{D}^{1/2}\mathcal{L}_n)\mathbf{D}^{-1}(\mathcal{L}_n\mathbf{D}^{1/2}\mathbf{x})$$

$$\gamma = \mathcal{L}_n\mathbf{D}^{1/2}\mathbf{x}$$

$$\mathbf{x}^T\mathcal{L}_r^T\mathcal{L}_r\mathbf{x} = \gamma^T\mathbf{D}^{-1}\gamma$$

$$\frac{\gamma^T\gamma}{d_{\max}} \leq \gamma^T\mathbf{D}^{-1}\gamma \leq \frac{\gamma^T\gamma}{d_{\min}}$$

$$(d_{\min}^{-1})\gamma^T\gamma$$

Random Walk Graph Laplacian

$$\boldsymbol{\gamma}^T\boldsymbol{\gamma} = \mathbf{x}^T\mathbf{D}^{1/2}\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T\mathbf{D}^{1/2}\mathbf{x}$$
$$= \boldsymbol{\beta}^T\boldsymbol{\Lambda}^2\boldsymbol{\beta} = \sum_k \tilde{\eta}_k^2\,\beta_k^2.$$

- We have a graph frequency interpretation of our Left Eigenvector Random-walk Graph Laplacian (LERaG) $(d_{\min}^{-1})\boldsymbol{\gamma}^T\boldsymbol{\gamma}$:

high frequencies of random walk graph Laplacian are suppressed to restore smooth signal x

- The proposed regularizer can be efficiently computed as:

$$(d_{\min}^{-1})\boldsymbol{\gamma}^T\boldsymbol{\gamma} = \mathbf{x}^T(d_{\min}^{-1})\mathbf{L}\mathbf{D}^{-1}\mathbf{L}\mathbf{x}$$

Only adjacency matrix is involved, no need to compute other matrix

Advantages of the Proposed Graph Laplacian

- Compared with combinatorial graph Laplacian

Our Laplacian is based on random walk graph Laplacian (normalized), therefore, it is insensitive to the degrees of graph vertices.

- Compared with normalized graph Laplacian

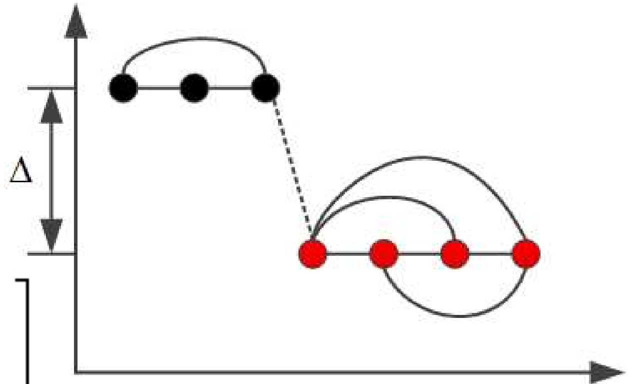
Our Laplacian can efficiently filter constant signals, thus is suitable for image filtering.

- Compared with doubly stochastic graph Laplacian

Our Laplacian can be computed simply.

Analysis of Ideal Piecewise Constant Signals

- 1D Piecewise constant (PWC) signal
- A full-connected graph is built



$$\mathbf{W} = \begin{bmatrix} \mathbf{A}_l & \mathbf{0}_{l\times(n-l)} \\ \mathbf{0}_{(n-l)\times l} & \mathbf{A}_{n-l} \end{bmatrix} \quad \mathcal{L}_n = \begin{bmatrix} \tilde{\mathbf{B}}_l & \mathbf{0}_{l\times(n-l)} \\ \mathbf{0}_{(n-l)\times l} & \tilde{\mathbf{B}}_{n-l} \end{bmatrix}$$

- The first eigenvector $\mathbf{v}_1 = \mathbf{D}^{1/2}\mathbf{1}$
- The second eigenvector $\mathbf{v}_2$

$$v_{2,i} = \begin{cases} 1/l(l-1)^{1/2} & \text{if } 1 \le i \le l \\ -1/(n-l)(n-l-1)^{1/2} & \text{if } l < i \le n \end{cases}$$

⇨ **PWC**

- We can see that $\mathbf{D}^{1/2}\mathbf{x} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2$

$$a_1 = \frac{c_1 l(l-1) + c_2(n-l)(n-l-1)}{(n-l)(n-l-1) + l(l-1)} \qquad a_2 = \frac{(c_1 - c_2)l(l-1)(n-l)(n-l-1)}{(n-l)(n-l-1) + l(l-1)}$$

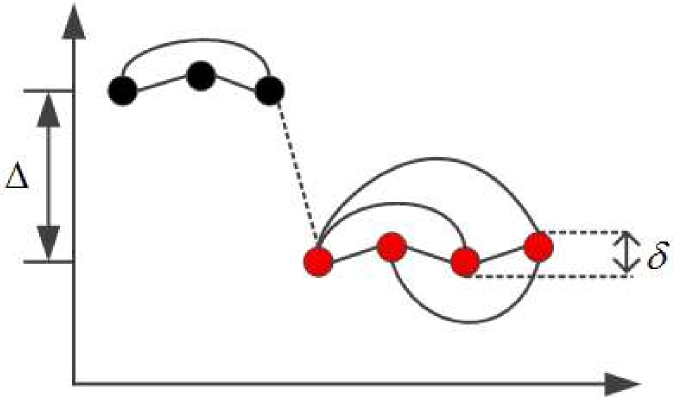
Analysis of Ideal Piecewise Constant Signals

Given an ideal two-piece PWC signal **x**, $\mathbf{D}^{1/2}\mathbf{x}$ can be represented exactly using the first two eigenvectors of L_n corresponding to eigenvalue 0, hence LERaG evaluates to 0.

- $\mathbf{D}^{1/2}\mathbf{x}$ is a ideal low-pass given eigenvectors of $\mathcal{L}_n$
- There is no penalty for LERaG.

Analysis of Piecewise Smooth Signals

- 1D piecewise smooth (PWS) signal:
- A full-connected graph is built



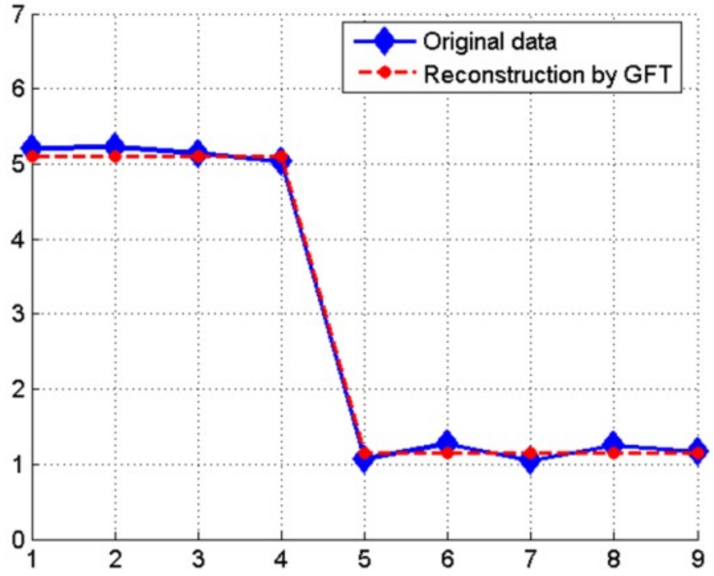
- The normalized graph Laplacian $\mathcal{L}_n$ is still block-diagonal
- The second eigenvector $\mathbf{v}_2$

$$v_{2,i} = \begin{cases} \frac{D_{i,i}^{1/2}}{\sum_{j=1}^{l} D_{j,j}} & \text{if } 1 \le i \le l \\ -\frac{D_{i,i}^{1/2}}{\sum_{j=l+1}^{n} D_{j,j}} & \text{if } 1 < i \le n \end{cases}$$

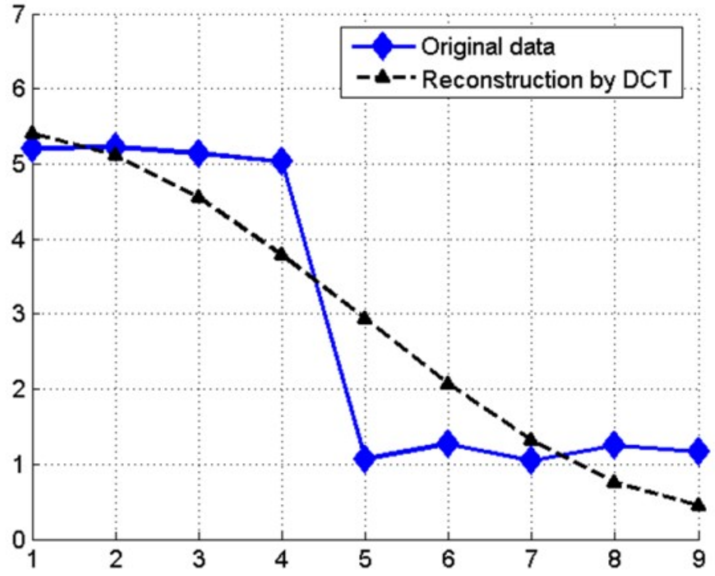
⇒ **Roughly PWS**

- $\mathbf{D}^{1/2}\mathbf{x}$ is also roughly PWS: $\mathbf{D}^{1/2}\mathbf{x} \approx a_1\mathbf{v}_1 + a_2\mathbf{v}_2$
- There is a small penalty of LERaG.

Analysis of Ideal Piecewise Smooth Signals



(a)



(b)

Soft Decoding via Priors Mixture

- The objective function

$$\arg\min_{\{\mathbf{x},\boldsymbol{\alpha}\}} \|\mathbf{x}-\boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 + \lambda_1\|\boldsymbol{\alpha}\|_0 + \lambda_2\mathbf{x}^T(d_{\min}^{-1})\mathbf{L}\mathbf{D}^{-1}\mathbf{L}\mathbf{x},$$
$$\text{s.t. } \mathbf{q}\mathbf{Q} \preceq \mathbf{T}\mathbf{M}\mathbf{x} \prec (\mathbf{q}+1)\mathbf{Q}$$

  - λ1 is fixed
  - We adaptively increase λ2 if q-bin indices q indicate the presence of high DCT frequencies in target **x**.

- Optimization
  - Laplacian prior provides an initial estimation;
  - Fix **x** and estimate **α**;
  - Fix **α** and estimate **x**.

Experimental Results

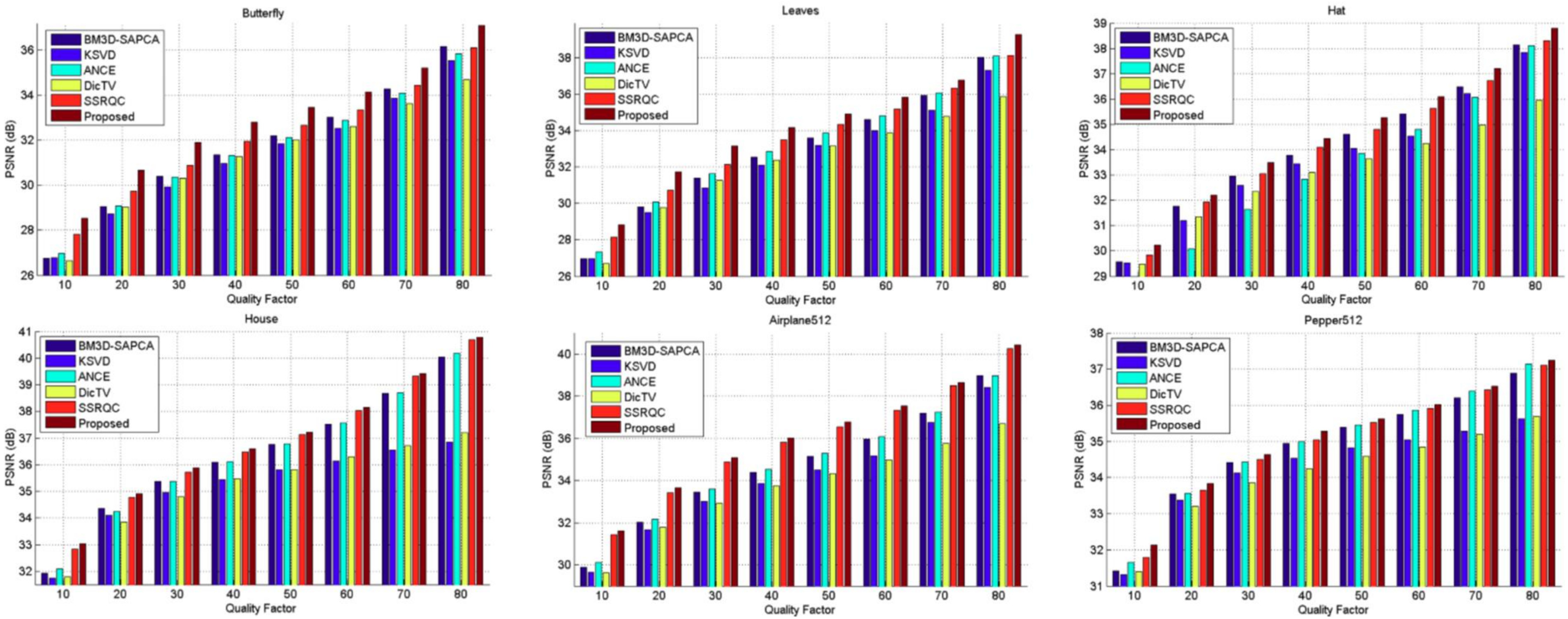
- Compared methods
  - BM3D: well-known denoising algorithm
  - KSVD: with a large enough over-complete dictionary (100x4000); our method uses a much smaller one (100x400).
  - ANCE: non-local self similarity **[Zhang et al. TIP14]**
  - DicTV: Sparsity + TV **[Chang et al, TSP15]**
  - SSRQC: Low rank + Quantization constraint **[Zhao et al. TCSVT16]**

PSNR and SSIM Evaluation

QUALITY COMPARISON WITH RESPECT TO PSNR (IN dB) AND SSIM AT QF = 40

Images	JPEG		BM3D [38]		KSVD [8]		ANCE [18]		DicTV [3]		SSRQC [20]		Ours	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Butterfly	29.97	0.9244	31.35	0.9555	31.57	0.9519	31.38	0.9548	31.22	0.9503	32.02	0.9619	**32.87**	**0.9627**
Leaves	30.67	0.9438	32.55	0.9749	33.04	0.9735	32.74	0.9728	32.45	0.9710	32.13	0.9741	**34.42**	**0.9803**
Hat	32.78	0.9022	33.89	0.9221	33.62	0.9149	33.69	0.9169	33.20	0.8988	34.10	0.9237	**34.46**	**0.9268**
Boat	33.42	0.9195	34.77	0.9406	34.28	0.9301	34.64	0.9362	26.08	0.7550	33.88	0.9306	**34.98**	**0.9402**
Bike	28.98	0.9131	29.96	0.9356	30.19	0.9323	30.31	0.9357	29.75	0.9154	30.35	0.9411	**31.14**	**0.9439**
House	35.07	0.8981	36.09	0.9013	36.05	0.9055	36.12	0.9048	35.17	0.8922	36.49	0.9072	**36.55**	**0.9071**
Flower	31.62	0.9112	32.81	0.9357	32.63	0.9271	32.67	0.9314	31.86	0.9084	33.02	0.9362	**33.37**	**0.9371**
Parrot	34.03	0.9291	34.92	0.9397	34.91	0.9371	35.02	0.9397	33.92	0.9227	35.11	0.9401	**35.32**	**0.9401**
Pepper512	34.21	0.8711	34.94	0.8767	34.89	0.8784	34.99	0.8803	34.24	0.8639	35.05	0.8795	**35.19**	**0.8811**
Fishboat512	32.76	0.8763	33.61	0.8868	33.36	0.8809	33.60	0.8861	32.53	0.8496	33.68	0.8859	**33.73**	**0.8871**
Lena512	35.12	0.9089	36.03	0.9171	35.82	0.9146	36.04	0.9177	34.85	0.8986	36.09	0.9187	**36.11**	**0.9191**
Airplane512	33.36	0.9253	34.38	0.9361	34.36	0.9341	34.53	0.9358	33.75	0.9134	35.81	0.9355	**36.07**	**0.9439**
Bike512	29.43	0.9069	30.47	0.9299	30.66	0.9258	30.71	0.9298	30.05	0.9043	32.26	0.9372	**32.55**	**0.9387**
Statue512	32.78	0.9067	33.61	0.9188	33.55	0.9149	33.55	0.9193	32.53	0.8806	34.88	0.9249	**34.95**	**0.9273**
Average	32.44	0.9097	33.52	0.9264	33.50	0.9229	33.57	0.9258	32.25	0.8945	33.91	0.9283	**34.41**	**0.9311**

QF-PSNR Evaluation



Subjective Quality Evaluation

(a) BM3D (23.91,0.8266)

(b) KSVD (24.55,0.8549)

(c) ANCE (24.34,0.8532)

(d) DicTV (23.42,0.8176)

(e) SSRQC (25.31,0.8764)

(f) Proposed (25.82,0.8861)

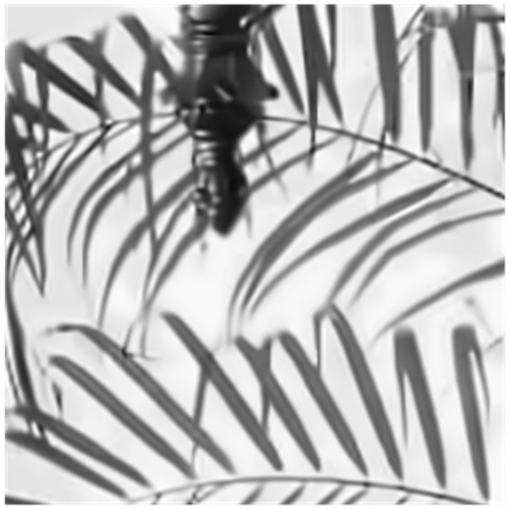
Subjective Quality Evaluation

(a) BM3D (23.78,0.8408)

(b) KSVD (24.39,0.8684)

(c) ANCE (24.18, 0.8551)

(d) DicTV (23.27,0.8245)



(e) SSRQC (25.01,0.8861)

(f) Proposed (25.57,0.8979)

Other Comparison

- Computation complexity comparison

TIME	BM3D	KSVD	ANCE	DicTV	SSRQC	Proposed
Average	373.35	209.71	307.43	39.53	70.32	143.73

- Comparison with other graph regularizers

Images	Combinatorial	Normalized	Doubly Stochastic	LERaG
Butterfly	25.42	24.70	25.15	**25.57**
Leaves	24.99	24.54	24.84	**25.17**
Hat	27.53	27.42	27.43	**27.56**
Boat	26.99	26.94	26.98	**26.99**
Bike	23.12	23.01	23.09	**23.17**
House	29.87	29.83	29.86	**29.89**
Flower	25.84	25.78	25.82	**25.87**
Parrot	27.97	27.95	27.97	**28.02**
Average	26.46	26.27	26.39	**26.53**

Conclusion

- We propose a new graph-signal smoothness prior based on left eigenvectors of the random walk graph Laplacian.
  - with desirable image filtering properties
  - can recover high DCT frequencies of piecewise smooth signals well
  - can be used in other image restoration or general GSP tasks

- We combine the Laplacian prior, sparsity prior and our new graph-signal smoothness prior into an efficient JPEG images soft decoding algorithm.

Thanks! Any Question?

xmliu.hit@gmail.com